



# BREVET D'INVENTION

CERTIFICAT D'UTILITÉ - CERTIFICAT D'ADDITION

## COPIE OFFICIELLE

Le Directeur général de l'Institut national de la propriété industrielle certifie que le document ci-annexé est la copie certifiée conforme d'une demande de titre de propriété industrielle déposée à l'Institut.

Fait à Paris, le **31 MARS 2000**

Pour le Directeur général de l'Institut  
national de la propriété industrielle  
Le Chef du Département des brevets

**PRIORITY DOCUMENT**  
SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH  
RULE 17.1(a) OR (b)

A handwritten signature in black ink, appearing to read 'M. Planche', enclosed within a large, loopy oval stroke.

Martine PLANCHE

## BEST AVAILABLE COPY

INSTITUT  
NATIONAL DE  
LA PROPRIÉTÉ  
INDUSTRIELLE

SIEGE  
26 bis, rue de Saint Petersburg  
75800 PARIS Cédex 08  
Téléphone : 01 53 04 53 04  
Télécopie : 01 42 93 59 30

**THIS PAGE BLANK (USPTO)**

26bis, rue de Saint-Petersbourg

75800 Paris Cedex 08

Téléphone: 01 53.04.53.04 Télécopie: 01.42.94.86.54

Code de la propriété intellectuelle-livre VI

REQUÊTE EN DÉLIVRANCE

<b>0</b>	<b>RESERVE A L'INPI</b>	
0-1	Date de remise des pièces	26/03/99
0-2	N° d'enregistrement national	9903920
0-3	Département de dépôt	99
0-4	Date de dépôt	26 MARS 1999
0-6	Titre de l'invention	
0-8	Etablissement du Rapport de Recherche	
0-9	Votre référence dossier	
<b>1</b>	<b>DEMANDEUR(s)</b>	
1-1	Nom Suivi par Adresse rue Adresse code postal et ville Pays Nationalité Forme juridique N° SIREN Code APE-NAF N° de téléphone N° de télécopie Courrier électronique	
	GEMPLUS BRUYERE Pierre Avenue du Pic de Bertagne Parc d'Activités de Gemenos 13881, GEMENOS France France SCA 349 711 200 321B 04.42.36.69.06. 04.42.36.63.43. nathalie.herail@gemplus.com	
<b>4</b>	Déclaration de PRIORITE ou REQUETE du bénéfice de la date de dépôt d'une demande antérieure	
	Etat	Date
	N° de la demande	
<b>6</b>	Documents et Fichiers joints	
6-1	Description	Fichier électronique
6-2	Revendications	gem652.doc
6-3	Listage de séquences	gem652.doc
6-4	Rapport de recherche	
<b>7</b>	Mode de paiement	
7-1	Numéro du compte client	
7-2	Remboursement à effectuer sur le compte n°	
<b>8</b>	REDEVANCES	
	Devise	Taux
062 Dépôt	FRF	250.00
063 Rapport de recherche (R.R.)	FRF	4 200.00
068 Revendication à partir de la 11ème	FRF	115.00
Total à acquitter	FRF	

10	Signature	
10-1	Signé par	NONNENMACHER Bernard Directeur de la Propriété Industrielle GEMPLUS

La loi n°78-17 du 6 janvier 1978 relative à l'informatique aux fichiers et aux libertés s'applique aux réponses faites à ce formulaire.  
Elle garantit un droit d'accès et de rectification pour les données vous concernant auprès de l'INPI.

Désignation de l'inventeur

Référence utilisateur: Référence système: N° d'enregistrement national:	GEM652 111111 729774,63400162 99 0 3 920
Titre de l'invention:	Procédés de contre-mesure dans un composant électronique mettant en oeuvre un algorithme de cryptographie à clé publique de type courbe elliptique
Le(s) soussigné(s):  Désigne(nt) en tant qu'inventeur(s): Inventeur 1	NONNENMACHER Bernard Directeur de la Propriété Industrielle GEMPLUS
Nom, Prénom: Adresse:	CORON, Jean-Sébastien 45 rue d'ULM F-75005 PARIS France
Signé par:  En qualité de: Date:	NONNENMACHER Bernard Directeur de la Propriété Industrielle GEMPLUS Directeur de la Propriété Industrielle 25 mars 1999

# DOCUMENT COMPORTANT DES MODIFICATIONS

PAGE(S) DE LA DESCRIPTION OU DES REVENDEICATIONS OU PLANCHE(S) DE DESSIN			R.M.*	DATE DE LA CORRESPONDANCE	TAMPON DATEUR DU CORRECTEUR
Modifiée(s)	Supprimée(s)	Ajoutée(s)			
21 et 22	23		RM	30.11.99	8 DEC. 1999 - B

Un changement apporté à la rédaction des revendications d'origine, sauf si celui-ci découle des dispositions de l'article R.612-36 du code de la Propriété Intellectuelle, est signalé par la mention «R.M.» (revendications modifiées).

PROCEDES DE CONTRE-MESURE DANS UN  
COMPOSANT ELECTRONIQUE METTANT EN ŒUVRE  
UN ALGORITHME DE CRYPTOGRAPHIE A CLE  
PUBLIQUE DE TYPE COURBE ELLIPTIQUE

La présente invention concerne un procédé de contre-mesure dans un composant électronique mettant en œuvre un algorithme de chiffrement à clé publique de type courbe elliptique

5

Dans le modèle classique de la cryptographie à clef secrète, deux personnes désirant communiquer par l'intermédiaire d'un canal non sécurisé doivent au préalable se mettre d'accord  
10 sur une clé secrète de chiffrement K. La fonction de chiffrement et la fonction de déchiffrement utilisent la même clef K. L'inconvénient du système de chiffrement à clé secrète est que ledit système requiert la  
15 communication préalable de la clé K entre les deux personnes par l'intermédiaire d'un canal sécurisé, avant qu'un quelconque message chiffré ne soit envoyé à travers le canal non sécurisé. Dans la pratique, il est généralement difficile  
20 de trouver un canal de communication parfaitement sécurisé, surtout si la distance séparant les deux personnes est importante. On entend par canal sécurisé un canal pour lequel il est impossible de connaître ou de modifier  
25 les informations qui transitent par ledit canal. Un tel canal sécurisé peut être réalisé par un câble reliant deux terminaux, possédés par les deux dites personnes.

Le concept de cryptographie à clef publique fut inventé par Whitfield DIFFIE et Martin HELLMAN en 1976. La cryptographie à clef publique permet de résoudre le problème de la distribution des

5 clefs à travers un canal non sécurisé. Le principe de la cryptographie à clef publique consiste à utiliser une paire de clefs, une clef publique de chiffrement et une clef privée de déchiffrement. Il doit être calculatoirement

10 infaisable de trouver la clef privée de déchiffrement à partir de la clef publique de chiffrement. Une personne A désirant communiquer une information à une personne B utilise la clef publique de chiffrement de la personne B. Seule

15 la personne B possède la clef privée associée à sa clef publique. Seule la personne B est donc capable de déchiffrer le message qui lui est adressé.

20 Un autre avantage de la cryptographie à clé publique sur la cryptographie à clé secrète est que la cryptographie à clef publique permet l'authentification par l'utilisation de signature électronique.

25 La première réalisation de schéma de chiffrement à clef publique fut mis au point en 1977 par Rivest, Shamir et Adleman, qui ont inventé le système de chiffrement RSA. La sécurité de RSA

30 repose sur la difficulté de factoriser un grand nombre qui est le produit de deux nombres premiers.

Depuis, de nombreux systèmes de chiffrement à clef publique ont été proposés, dont la sécurité repose sur différents problèmes calculatoires : (cette liste n'est pas exhaustive).

5

- Sac à dos de Merckle-Hellman :

Ce système de chiffrement est basé sur la difficulté du problème de la somme de sous-ensembles.

10

- McEliece :

Ce système de chiffrement est basé sur la théorie des codes algébriques. Il est basé sur le problème du décodage de codes linéaires.

15

- ElGamal :

Ce système de chiffrement est basé sur la difficulté du logarithme discret dans un corps fini.

20

- Courbes elliptiques :

Le système de chiffrement à courbe elliptique constitue une modification de systèmes cryptographiques existant pour les appliquer au domaine des courbes elliptiques.

25

L'utilisation de courbes elliptiques dans des systèmes cryptographiques fut proposé indépendamment par Victor Miller et Neal Koblitz en 1985. Les applications réelles des courbes elliptiques ont été envisagées au début des années 1990.

30

L'avantage de cryptosystèmes à base de courbe elliptique est qu'ils fournissent une sécurité équivalente aux autres cryptosystèmes mais avec des tailles de clef moindres. Ce gain en taille  
 5 de clé implique une diminution des besoins en mémoire et une réduction des temps de calcul, ce qui rend l'utilisation des courbes elliptiques particulièrement adaptées pour des applications de type carte à puce.

10

Une courbe elliptique sur un corps fini  $GF(q^n)$  ( $q$  étant un nombre premier et  $n$  un entier) est l'ensemble des points  $(x,y)$  avec  $x$  l'abscisse et  $y$  l'ordonnée appartenant à  $GF(q^n)$   
 15 solution de l'équation :

$$y^2 = x^3 + ax + b$$

si  $q$  est supérieur ou égal à 3 et

20

$$y^2 + x*y = x^3 + a*x^2 + b$$

si  $q=2$ .

25

Les deux classes de courbes élliptiques les plus utilisées en cryptographie sont les classes suivantes :

1) Courbes définies sur le corps fini  $GF(p)$   
 30 (ensemble des entiers modulo  $p$ ,  $p$  étant un nombre premier) ayant pour équation:

$$y^2 = x^3 + ax + b$$

2) Courbes élliptiques sur le corps fini  $GF(2^n)$  ayant pour équation  $y^2+xy=x^3+ax^2+b$

5 Pour chacune de ces deux classes de courbes, on définit une opération d'addition de points: étant donné deux points  $P$  et  $Q$ , la somme  $R=P+Q$  est un point de la courbe, dont les coordonnées s'expriment à l'aide des  
10 coordonnées des points  $P$  et  $Q$  suivant des formules dont l'expression est donnée dans l'ouvrage « Elliptic Curve public key cryptosystem » par Alfred J. Menezes.

Cette opération d'addition permet de définir une  
15 opération de multiplication scalaire: étant donné un point  $P$  appartenant à une courbe élliptique et un entier  $d$ , le résultat de la multiplication scalaire de  $P$  par un point  $d$  tel que  $Q=d.P=P+PP....+P$   $d$  fois.

20

La sécurité des algorithmes de cryptographie sur courbes élliptiques est basée sur la difficulté  
25 du logarithme discret sur courbes élliptiques, ledit problème consistant à partir de deux points  $Q$  et  $P$  appartenant à une courbe élliptique  $E$ , de trouver, s'il existe, un entier  $x$  tel que  $Q=x.P$

30

Il existe de nombreux algorithmes cryptographiques basés sur le problème du logarithme discret.

Ces algorithmes sont facilement transposables aux courbes elliptiques. Ainsi, il est possible de mettre en œuvre des algorithmes assurant l'authentification, la confidentialité, le  
 5 contrôle d'intégrité et l'échange de clé.

Un point commun à la plupart des algorithmes cryptographiques basés sur les courbes elliptiques est qu'ils comprennent comme  
 10 paramètre une courbe elliptique définie sur un corps fini et un point  $P$  appartenant à cette courbe elliptique. La clé privée est un entier  $d$  choisi aléatoirement. La clef publique est un point  $Q$  de la courbe tel que  $Q = d.P$ . Ces  
 15 algorithmes cryptographiques font généralement intervenir une multiplication scalaire dans le calcul d'un point  $R = d.T$  où  $d$  est la clef secrète.

20 Dans ce paragraphe, on décrit un algorithme de chiffrement à base de courbe elliptique. Ce schéma est analogue au schéma de chiffrement d'El Gamal. Un message  $m$  est chiffré de la manière suivante :

25

Le chiffeur choisit un entier  $k$  aléatoirement et calcule les points  $k.P = (x_1, y_1)$  et  $k.Q = (x_2, y_2)$  de la courbe, et l'entier  $c = x_2 + m$ . Le chiffré de  $m$  est le triplet  $(x_1, y_1, c)$ .

30 Le déchiffreur qui possède  $d$  déchiffre  $m$  en calculant :

$$(x'_2, y'_2) = d(x_1, y_1) \text{ et } m = c - x'_2$$

Pour réaliser les multiplications scalaires nécessaires dans les procédés de calcul décrits précédemment, plusieurs algorithmes existent :

- 5     Algorithme " double and add " ;
- Algorithme " addition-soustraction "
- Algorithme avec chaînes d'addition ;
- Algorithme avec fenêtre ;
- Algorithme avec représentation signée ;

10

Cette liste n'est pas exhaustive. L'algorithme le plus simple et le plus utilisé est l'algorithme " double and add ". L'algorithme " double and add " prend en entrée un point  $P$  appartenant à une courbe elliptique donnée et un entier  $d$ . L'entier  $d$  est noté  $d = (d(t), d(t-1), \dots, d(0))$ , où  $(d(t), d(t-1), \dots, d(0))$  est la représentation binaire de  $d$ , avec  $d(t)$  le bit de poids fort et  $d(0)$  le bit de poids faible.

15

20 L'algorithme retourne en sortie le point  $Q = d.P$ .

L'algorithme " double and add " comporte les 3 étapes suivantes :

- 25   1) Initialiser le point  $Q$  avec la valeur  $P$
- 2) Pour  $i$  allant de  $t-1$  à  $0$  exécuter :
  - 2a)       Remplacer  $Q$  par  $2Q$
  - 2b)       Si  $d(i)=1$  remplacer  $Q$  par  $Q+P$
- 3) Retourner  $Q$ .

30

Il est apparu que l'implémentation sur carte à puce d'un algorithme de chiffrement à clé publique du type courbe elliptique était vulnérable à des attaques consistant en une  
5 analyse différentielle de consommation de courant permettant de retrouver la clé privée de déchiffrement. Ces attaques sont appelées attaques DPA, acronyme pour Differential Power Analysis. Le principe de ces attaques DPA repose  
10 sur le fait que la consommation de courant du microprocesseur exécutant des instructions varie selon la donnée manipulée.

En particulier, lorsqu'une instruction manipule  
15 une donnée dont un bit particulier est constant, la valeur des autres bits pouvant varier, l'analyse de la consommation de courant liée à l'instruction montre que la consommation moyenne de l'instruction n'est pas la même suivant que  
20 le bit particulier prend la valeur 0 ou 1. L'attaque de type DPA permet donc d'obtenir des informations supplémentaires sur les données intermédiaires manipulées par le microprocesseur de la carte lors de l'exécution d'un algorithme  
25 cryptographique. Ces informations supplémentaires peuvent dans certain cas permettre de révéler les paramètres privés de l'algorithme de déchiffrement, rendant le système cryptographique non sûr.

Dans la suite de ce document on décrit un procédé d'attaque DPA sur un algorithme de type courbe elliptique réalisant une opération du type multiplication scalaire d'un point  $P$  par un entier  $d$ , l'entier  $d$  étant la clé secrète. Cette attaque permet de révéler directement la clé secrète  $d$ . Elle compromet donc gravement la sécurité de l'implémentation de courbes elliptiques sur une carte à puce.

10

La première étape de l'attaque est l'enregistrement de la consommation de courant correspondant à l'exécution de l'algorithme "double and add" décrit précédemment pour  $N$  points distincts  $P(1), \dots, P(N)$ . Dans un algorithme à base de courbes elliptiques, le microprocesseur de la carte à puce va effectuer  $N$  multiplications scalaires  $d.P(1), \dots, d.P(N)$ .

Pour la clarté de la description de l'attaque, on commence par décrire une méthode permettant d'obtenir la valeur du bit  $d(t-1)$  de la clé secrète  $d$ , où  $(d(t), d(t-1), \dots, d(0))$  est la représentation binaire de  $d$ , avec  $d(t)$  le bit de poids fort et  $d(0)$  le bit de poids faible. On donne ensuite la description d'un algorithme qui permet de retrouver la valeur de  $d$ .

On groupe les points  $P(1)$  à  $P(N)$  suivant la valeur du dernier bit de l'abscisse de  $4.P$ , où  $P$  désigne un des points  $P(1)$  à  $P(N)$ . Le premier groupe est constitué des points  $P$  tels que le dernier bit de l'abscisse de  $4.P$  est égal à 1.

Le second groupe est constitué des points  $P$  tels que le dernier bit de l'abscisse de  $4.P$  est égal à 0. On calcule la moyenne des consommations de courant correspondant à chacun des deux groupes, et on calcule la courbe de différence entre ces deux moyennes.

Si le bit  $d(t-1)$  de  $d$  est égal à 0, alors l'algorithme de multiplication scalaire précédemment décrit calcule et met en mémoire la valeur de  $4.P$ . Cela signifie que lors de l'exécution de l'algorithme dans une carte à puce, le microprocesseur de la carte va effectivement calculer  $4.P$ . Dans ce cas, dans le premier groupe de message le dernier bit de la donnée manipulée par le microprocesseur est toujours à 1, et dans le deuxième groupe de message le dernier bit de la donnée manipulée est toujours à 0. La moyenne des consommations de courant correspondant à chaque groupe est donc différente. Il apparaît donc dans la courbe de différence entre les 2 moyennes un pic de différentiel de consommation de courant.

Si au contraire le bit  $d(t-1)$  de  $d$  est égal à 1, l'algorithme d'exponentiation décrit précédemment ne calcule pas le point  $4.P$ . Lors de l'exécution de l'algorithme par la carte à puce, le microprocesseur ne manipule donc jamais la donnée  $4.P$ . Il n'apparaît donc pas de pic de différentiel de consommation.

Cette méthode permet donc de déterminer la valeur du bit  $d(t-1)$  de  $d$ .

5 L'algorithme décrit dans le paragraphe suivant est une généralisation de l'algorithme précédant. Il permet de déterminer la valeur de la clé secrète  $d$  :

10 On définit l'entrée par  $N$  points notés  $P(1)$  à  $P(N)$  correspondant à  $N$  calculs réalisés par la carte à puce et la sortie par un entier  $h$ .

Ledit algorithme s'effectue de la manière suivante en trois étapes.

15

1) Exécuter  $h=1$  ;

2) Pour  $i$  allant de  $t-1$  à 1, exécuter :

20 2)1) Classer les points  $P(1)$  à  $P(N)$  suivant la valeur du dernier bit de l'abscisse de  $(4 \cdot h) \cdot P$  ;

2)2) Calculer la moyenne de consommation de courant pour chacun des deux groupes ;

2)3) Calculer la différence entre les 2 moyennes ;

25 2)4) Si la différence fait apparaître un pic de différentiel de consommation, faire  $h=h \cdot 2$  ; sinon faire  $h=h \cdot 2 + 1$  ;

3) Retourner  $h$ .

30 L'algorithme précédent fournit un entier  $h$  tel que  $d=2 \cdot h$  ou  $d=2 \cdot h + 1$ . Pour obtenir la valeur de  $d$ , il suffit ensuite de tester les deux hypothèses possibles.

L'attaque de type DPA décrite permet donc de retrouver la clé privée  $d$ .

5 Le procédé de l'invention consiste en l'élaboration de trois contre-mesures permettant de se prémunir contre l'attaque DPA précédemment décrite.

10 Le procédé de la première contre-mesure consiste à calculer à partir de la clé privée  $d$  et du nombre de points  $n$  de la courbe elliptique un nouvel entier de déchiffrement  $d'$ , tel que le déchiffrement d'un message chiffré quelconque avec  $d'$  donne le même résultat qu'avec  $d$ .

15

Dans le cas d'un algorithme cryptographique basé sur l'utilisation de courbes elliptiques réalisant l'opération  $Q=d.P$  où  $d$  est la clé privée et  $P$  un point de la courbe, le calcul de  
20  $Q=d.P$  est remplacé par le procédé suivant en quatre étapes:

1) Détermination d'un paramètre de sécurité  $s$ , dans la pratique on peut prendre  $s$  voisin de 30.

25

2) Tirage d'un nombre aléatoire  $k$  compris entre 0 et  $2^s$ ;

3) Calcul de l'entier  $d'=d+k*n$ ;

30

4) Calcul de  $Q=d'.P$ .

Le procédé de la première contre-mesure comprend deux variantes qui concernent la mise à jour de l'entier  $d'$ . La première variante consiste en ce qu'un nouvel entier de déchiffrement  $d'$  est  
 5 calculé à chaque nouvelle exécution de l'algorithme de déchiffrement, selon le procédé décrit précédemment. La seconde variante consiste en ce qu'un compteur est incrémenté à chaque nouvelle exécution de l'algorithme de  
 10 déchiffrement. Lorsque ce compteur atteint une valeur fixée  $T$ , un nouvel entier de déchiffrement  $d'$  est calculé selon le procédé décrit précédemment, et le compteur est remis à zéro. Dans la pratique, on peut prendre  $T=16$ .

15

Le procédé de la première contre-mesure rend donc l'attaque DPA précédemment décrite impossible en changeant d'entier  $d$  de déchiffrement.

20

Le procédé de la deuxième contre-mesure s'applique à la première classe de courbes précédemment décrites, c'est à dire les courbes définies sur le corps fini  $GF(p)$  ayant pour  
 25 équation  $y^2 = x^3 + ax + b$ . Le procédé de la deuxième contre-mesure consiste à utiliser un module de calcul aléatoire à chaque nouvelle exécution. Ce module aléatoire est de la forme  $p' = p * r$  où  $r$  est un entier aléatoire. L'opération de  
 30 multiplication scalaire  $Q = d.p$  réalisée dans un algorithme à base de courbe elliptique s'effectue alors selon le procédé suivant en cinq étapes:

- 1) Détermination d'un paramètre de sécurité  $s$ ;  
dans la pratique, on peut prendre  $s$  voisin du  
nombre 60;
- 5 2) Tirage du nombre aléatoire  $r$  dont la  
représentation binaire fait  $s$  bits;
- 3) Calcul de  $p' = p * r$ ;
- 4) Exécuter l'opération de multiplication  
scalaire  $Q = d.P$ , les opérations étant effectuées  
10 modulo  $p'$ ;
- 5) Effectuer l'opération de réduction modulo  $p$   
des coordonnées du point  $Q$ .

Le procédé de la seconde contre-mesure comprend  
15 deux variantes qui concernent la mise à jour de  
l'entier  $r$ . La première variante consiste en ce  
qu'un nouvel entier  $r$  est calculé à chaque  
nouvelle exécution de l'algorithme de  
déchiffrement, selon le procédé décrit  
20 précédemment. La seconde variante consiste en ce  
qu'un compteur est incrémenté à chaque nouvelle  
exécution de l'algorithme de déchiffrement.  
Lorsque ce compteur atteint une valeur fixée  $T$ ,  
un nouvel entier  $r$  est calculé selon le procédé  
25 décrit précédemment, et le compteur est remis à  
zéro.. Dans la pratique, on peut prendre  $T+16$ .

Le procédé de la troisième contre-mesure  
consiste à « masquer » le point  $P$  sur lequel on  
30 veut appliquer l'algorithme de multiplication  
scalaire en lui ajoutant un point aléatoire  $R$ .

Le procédé de multiplication scalaire d'un point P par un entier d suivant  $Q=d.P$  comprend les cinq étapes suivantes:

- 5 1) Tirage d'un point aléatoire R sur la courbe;
  - 2) Calcul de  $P'=P+R$ ;
  - 3) Opération de multiplication scalaire  $Q'=d.P'$ ;
  - 10 4) Opération de multiplication scalaire  $S=d.R$ ;
  - 5) Calcul de  $Q=Q'-S$ .
- 15 Le procédé de la troisième contre-mesure comprend trois variantes. la première variante consiste en ce qu'un compteur est incrémenté à chaque nouvelle exécution de l'algorithme de déchiffrement. Lors de la première exécution de
- 20 l'algorithme de déchiffrement, l'algorithme est exécuté suivant le procédé en cinq étapes décrit précédemment. Tant que le compteur n'a pas atteint la valeur limite T, les étapes 1 et 4 du procédé décrit précédemment ne sont pas
- 25 exécutées, les points R et S gardant les valeurs prises lors de l'exécution précédente. Lorsque le compteur atteint la valeur limite T, l'algorithme de déchiffrement s »effectue
- 30 suivant le procédé décrit précédemment en cinq étapes, et le compteur est remis à zéro. Dans la pratique, on peut prendre  $T=16$ .

La deuxième variante consiste en ce que la carte possède initialement en mémoire deux points de la courbe elliptique tels que  $S=d.R$ . Les étapes 1 et 4 de l'algorithme de déchiffrement 5 précédent sont remplacées par les étapes 1' et 4' suivantes:

- 1') Remplacer R par 2.R:
- 10 4') Remplacer S par 2.S.

La troisième variante consiste en une modification de la deuxième variante caractérisée en ce qu'un compteur est incrémenté 15 à chaque nouvelle exécution de l'algorithme de déchiffrement. Lors de la première exécution de l'algorithme de déchiffrement, l'algorithme est exécuté suivant le procédé en cinq étapes de la deuxième variante décrit précédemment. Tant que 20 le compteur n'a pas atteint une valeur limite T, les étapes 1' et 4' du procédé décrit précédemment ne sont pas exécutées, les points R et S gardant les valeurs prises lors de l'exécution précédente. Lorsque le compteur 25 atteint une valeur limite T, l'algorithme de déchiffrement s'effectue suivant le procédé précédemment décrit en cinq étapes, et le compteur est remis à zéro. Dans la pratique, on peut prendre  $T=16$ .

L'application des trois procédés de contre-mesure précédents permet de protéger tout l'algorithme cryptographique basé sur les courbes elliptiques contre l'attaque DPA  
5 précédemment décrit. Les trois contre-mesures présentées sont de plus compatibles entre elles: il est possible d'appliquer à l'algorithme de déchiffrement RSA une, deux ou trois des contre-mesures décrites.

# REVENDECATIONS

1- Procédé de contre-mesure dans un composant électronique mettant en oeuvre un algorithme de cryptographie à clé publique basé sur l'utilisation des courbes elliptiques consistant à calculer à partir de la clé privée  $d$  et du nombre de points  $n$  de ladite courbe elliptique un nouvel entier de déchiffrement  $d'$  tel que le déchiffrement d'un message chiffré quelconque, au moyen d'un algorithme de déchiffrement, avec  $d'$  donne le même résultat qu'avec  $d$ , en réalisant l'opération  $Q=d*P$ ,  $P$  étant un point de la courbe, procédé caractérisé en ce qu'il comprend quatre étapes:

15 1) Détermination d'un paramètre de sécurité  $s$ , dans la pratique on peut prendre  $s$  voisin de 30;  
2) Tirage d'un nombre aléatoire  $k$  compris entre 0 et  $2^s$ ;

20 3) Calcul de l'entier  $d'=d+k*n$ ;

4) Calcul de  $Q=d'.P$ .

2- Procédé de contre-mesure selon la revendication 1 caractérisé en ce qu'une première variante consiste en ce qu'un nouvel entier de déchiffrement  $d'$  est calculé à chaque nouvelle exécution de l'algorithme de déchiffrement.

3- Procédé de contre-mesure selon la revendication 1 caractérisé en ce qu'une seconde variante consiste en ce qu'un compteur est incrémenté à chaque nouvelle exécution de l'algorithme de déchiffrement jusqu'à atteindre une valeur fixée T.

4- Procédé de contre-mesure selon la revendication 3 caractérisé en ce qu'une fois la valeur T atteinte, un nouvel entier de chiffrement est calculé selon le procédé de la revendication 1 et le compteur est remis à zéro.

5- Procédé de contre-mesure selon la revendication 3 caractérisé la valeur T est égale à l'entier seize.

6- Procédé de contre-mesure dans un composant électronique mettant en oeuvre un algorithme de cryptographie à clé publique basé sur l'utilisation des courbes elliptiques définies sur un corps fini  $GF(p)$ ,  $p$  étant un nombre premier, ayant pour équation  $y^2 = x^3 + ax + b$ , consistant à utiliser un module de calcul aléatoire à chaque nouvelle exécution de la forme  $p' = p * r$  où  $r$  est un entier aléatoire et présentant un point  $P$  caractérisé en ce que ledit procédé exécute l'opération de multiplication scalaire en cinq étapes:

- 1) Détermination d'un paramètre de sécurité  $s$ ;  
dans la pratique, on peut prendre  $s$  voisin du  
nombre 60;
  - 2) Tirage du nombre aléatoire  $r$  dont la  
5 représentation binaire fait  $s$  bits;
  - 3) Calcul de  $p' = p * r$ ;
  - 4) Exécuter l'opération de multiplication  
scalaire  $Q = d.P$ , les opérations étant effectuées  
modulo  $p'$ ;
  - 10 5) Effectuer l'opération de réduction modulo  $p$   
des coordonnées du point  $Q$ .
- 7- Procédé de contre-mesure selon la  
revendication 6 caractérisé en ce qu'un nouvel  
15 entier est calculé à chaque nouvelle exécution  
de l'algorithme de déchiffrement.
- 8- Procédé de contre-mesure selon la  
revendication 6 caractérisé en ce qu'un compteur  
20 est incrémenté à chaque nouvelle exécution de  
l'algorithme de déchiffrement.
- 9- Procédé de contre-mesure selon la  
revendication 8 caractérisé en ce que le  
25 compteur est remis à zéro lorsqu'il a atteint  
une valeur  $T$ .
- 10- Procédé de contre-mesure selon la  
revendication 8 ou la revendication 9  
30 caractérisé en ce que la valeur  $T$  est égale à  
seize.

11. Procédé de contre-mesure dans un composant électronique mettant en oeuvre un algorithme de cryptographie à clé publique basé sur  
5 l'utilisation des courbes elliptiques consistant à calculer à partir de la clé privée  $d$  et du nombre de points  $n$  de ladite courbe elliptique un nouvel entier de déchiffrement  $d'$  tel que le déchiffrement d'un message chiffré quelconque,  
10 aumoyen d'un algorithme de déchiffrement, avec  $d'$  donne le même résultat qu'avec  $d$ , en réalisant l'opération  $Q=d \cdot P$ ,  $P$  étant un point de la courbe sur lequel est appliqué l'algorithme de multiplication scalaire en lui ajoutant un point  
15 aléatoire  $R$  par un entier  $d$  suivant l'équation  $Q=d \cdot P$ , procédé caractérisé en ce qu'il comprend cinq étapes suivantes:

- 20 1) Tirage d'un point aléatoire  $R$  sur la courbe;
- 2) Calcul de  $P' = P + R$ ;
- 3) Opération de multiplication scalaire  $Q' = d \cdot P'$ ;
- 25 4) Opération de multiplication scalaire  $S = d \cdot R$ ;
- 5) Calcul de  $Q = Q' - S$ .

12- Procédé de contre-mesure selon la  
30 revendication 12 caractérisé en ce qu'un compteur est incrémenté à chaque nouvelle

exécution de l'algorithme de déchiffrement jusqu'à une valeur T.

13- Procédé de contre-mesure selon la  
5 revendication 12 caractérisé en ce que le  
compteur est remis à zéro une fois atteint la  
valeur T.

14- Procédé de contre-mesure selon la  
10 revendication 12 caractérisé en ce qu'un  
compteur est incrémenté à chaque nouvelle  
exécution de l'algorithme de déchiffrement  
jusqu'à une valeur T.

15 15- Procédé de contre-mesure selon la  
revendication 11 caractérisé en ce que la courbe  
elliptique possède en mémoire deux points tels  
que  $S=d \cdot R$ , les étapes 1 et 4 étant alors  
remplacé par les étapes 1' et 4':

20

1') Remplacer R par 2.R:

4') Remplacer S par 2.S.

25 16- Procédé de contre-mesure selon la  
revendication 15 caractérisé en ce qu'un  
compteur est incrémenté à chaque nouvelle  
exécution de l'algorithme de déchiffrement  
jusqu'à une valeur T.

30

17- Procédé de contre-mesure selon la revendication 15 caractérisé en ce qu'un compteur est incrémenté à chaque nouvelle exécution de l'algorithme de déchiffrement  
5 jusqu'à une valeur T.

11- Procédé de contre-mesure dans un  
5 composant électronique mettant en oeuvre un  
algorithme de cryptographie à clé publique  
basé sur l'utilisation des courbes  
elliptiques consistant à calculer à partir  
de la clé privée  $d$  et du nombre de points  $n$   
10 de ladite courbe elliptique un nouvel entier  
de déchiffrement  $d'$  tel que le déchiffrement  
d'un message chiffré quelconque, au moyen  
d'un algorithme de déchiffrement, avec  $d'$   
donne le même résultat qu'avec  $d$ , en  
15 réalisant l'opération  $Q=d*P$ ,  $P$  étant un  
point de la courbe sur lequel est appliqué  
l'algorithme de multiplication scalaire en  
lui ajoutant un point aléatoire  $R$  par un  
entier  $d$  suivant l'équation  $Q=d*P$ , procédé  
20 caractérisé en ce qu'il comprend cinq étapes  
suivantes:

- 1) Tirage d'un point aléatoire  $R$  sur la courbe;
- 2) Calcul de  $P'=P+R$ ;
- 25 3) Opération de multiplication scalaire  $Q'=d.P'$ ;
- 4) Opération de multiplication scalaire  $S=d.R$ ;
- 5) Calcul de  $Q=Q'-S$ .

30 12- Procédé de contre-mesure dans un  
composant électronique selon la  
revendication 11 caractérisé en ce que ledit  
composant comprend un compteur incrémenté à  
35 chaque nouvelle exécution de l'algorithme de  
déchiffrement jusqu'à une valeur  $T$ .

- 5        13- Procédé de contre-mesure selon la  
revendication 12 caractérisé en ce que le  
compteur est remis à zéro une fois atteint  
la valeur T.
- 10       14- Procédé de contre-mesure selon la  
revendication 11 caractérisé en ce que la  
courbe elliptique possède en mémoire deux  
points tels que  $S=d \cdot R$ , les étapes 1 et 4  
étant alors remplacées par les étapes 1' et  
15       4' :
- 1' ) Remplacer R par 2.R:
- 4' ) Remplacer S par 2.S.
- 20       15- Procédé de contre-mesure selon la  
revendication 5 caractérisé en ce qu'un  
compteur est incrémenté à chaque nouvelle  
exécution de l'algorithme de déchiffrement  
25       jusqu'à une valeur T.
- 30       16- Procédé de contre-mesure selon la  
revendication 15 caractérisé en ce qu'un  
compteur est incrémenté à chaque nouvelle  
exécution de l'algorithme de déchiffrement  
jusqu'à une valeur T.

**THIS PAGE BLANK (USPTO)**

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**

**THIS PAGE BLANK (USPTO)**